Supporting collaborative multi-criteria evaluation: the VIP Analysis plug-in for Decision Deck

João N. Clímaco¹², João A. Costa¹, Luís C. Dias¹², Paulo Melo¹²

¹ INESC Coimbra, Rua Antero de Quental, 199, 3000-033 Coimbra, PORTUGAL
{jclimaco, jacosta, ldias, pmelo}@inescc.pt
² Faculty of Economics, University of Coimbra, Av. Dias da Silva 165, 3004-512 Coimbra, PORTUGAL.

Abstract: This paper presents the VIP Analysis plug-in of Decision Deck 1.1, a platform that hosts different evaluation methods to support decision makers in the collaborative evaluation of alternatives in a multi-criteria and multi-experts setting. VIP Analysis is a tool for aggregation of multicriteria performances by means of an additive value function under imprecise information. It allows conducting a multicriteria analysis for selecting an alternative when the decision makers are not able to (or do not wish to) fix precise values for the importance parameters. These parameters are seen as variables that may take several values subject to constraints. VIP Analysis incorporates different methods to support the progressive reduction of the number of alternatives, introducing a concept of tolerance that lets decision makers use some of the methods in a more flexible manner. When compared with the original standalone VIP Analysis programmed in the late 1990s the main innovation of the VIP Analysis plug-in is to allow several users working on the same problem under different roles: coordinator, evaluator, and decision-maker, thus defining a workflow process and enabling concurrent and remote access to the data over a network. By being included in the Decision Deck platform, VIP Analysis is now integrated with other decision aiding methods within a coherent interface. A final advantage is that the platform is open-source, which facilitates customization and collaborative improvement of the software.

Key words: group decisions, collaborative decision making, multi-criteria decision aiding (MCDA), Decision Deck.
Introduction

VIP Analysis (Dias & Clímaco, 2000) is a multi-criteria decision support tool to evaluate a discrete set of alternatives in choice problems. Based on the additive model for the aggregation of value functions (Keeney & Raiffa, 1976), its main characteristic is that it does not require a Decision Maker (DM) to indicate precise values for the trade-offs between different criteria. Rather, it can accept imprecise / partial information (namely intervals and linear constraints) on these values. VIP Analysis may be used to discover robust conclusions, i.e. those that hold for every accepted combination of the parameters, and to identify which results are more affected by the imprecision in the parameter values, i.e. the variability of the results.

In its original form, VIP Analysis can be a useful tool for a single user, which can be a single DM, or someone using the tool to support the discussion of a group gathered at the same location, as in a decision conference (Phillips & Phillips, 1993). In this work, we introduce a new version that allows a group of actors engaged in a decision process to collaborate over the internet, hence not needing to meet at the same location, nor at the same time. We have used the Decision Deck (D²) platform (Bisdorff et al., 2007; Decision Deck, 2008a), an open-source software offering a generic multi-criteria evaluation structure where new methods can be plugged-in. The platform allows a workflow-like organization of the decision process, where multiple users can participate in coordinating roles, specialist evaluation roles, or as decision analysts dealing with the more political aspects of the problem (e.g., defining the importance of the evaluation criteria).

According to Bergoggi (2003), Internet Multiattribute Group Decision Making (IMGDM) is a new approach to distribute group decision making intending to arrive at a consensus preference ranking of all the alternatives, by aggregating the group members preferences. However, not only DMs may find it hard to provide precise figures about their preferences, but also these preferences may change as the decision making process evolves. The procedures used to elicit the values of the trade-offs (weights, or scaling constants) may require more time and patience than some DMs can spare. Moreover, in group decision situations the opinions and preferences of the DMs diverge frequently.

We argue that the use of incomplete information regarding the trade-offs required by VIP Analysis helps to overcome this situation. As it is emphasized by Sen (1999), “preference formation through social interaction is a major subject of interest […]. It is also important to recognize that agreed social arrangements and adequate public policies do not require that there be a unique “social ordering” that completely ranks all the alternative social possibilities. Partial agreements still separate out acceptable options and a workable solution can be based
on the contingent acceptance of particular provisions, without demanding complete social unanimity [...]

VIP Analysis is based on a similar point of view. The requirement of complete information on the trade-offs between criteria is relaxed, allowing to proceed with variable interdependent parameters. This means that instead of requiring precise values for the weights it requires just intervals, and/or other linear constraints on these values. The constraints usually stem from imprecise answers from the DM or from holistic judgments about the alternatives that the DM is able to compare.

So, the DMs just fix those constraints on the scaling constants they feel confident to do. Consequently, either in individual or in group settings, VIP Analysis is a constructive approach looking for robust conclusions, trying to identify the “most satisfactory” solution rather than preference rankings of all the alternatives.

Another key and controversial issue is the discussion on the effectiveness of IMGDM in comparison with the traditional face to face settings.

Communication support for the aggregation of preferences from different DMs must play a central role in IMGDM (Tung & Turban, 1999). It must be taken into account that ambiguity of computer-mediated communication together with the dynamics of the process may lead to hinder the possibility of obtaining good group consensus. Dialogue requirements should be adequate for building effective asynchronous communication settings.

Wierzbicki (2008) emphasises: “if we accept the point that we are living in the time of changing civilization eras, and conceptual change is one of the main ingredients of the civilization change, up to the formation of the new episteme, then the need of new concepts and approaches, even new hermeneutical horizons also within group decisions and negotiation theory is evident”. Furthermore, it is specified by the author that “among other dimensions, distributed decision making may involve net technologies”. Indeed, the Internet and the Computer are omnipresent in today’s organizations. Despite all the barriers that exist in computer-mediated communication (Thompson, 2001), DMs – older or younger – are increasingly accustomed to using the Internet to communicate or to work collaboratively. There are many benefits in face-to-face meetings for model building (Phillips & Phillips, 1993), particularly concerning interpersonal relations, which make it advisable to use IMGDM mostly for groups whose members already know each other. IMGDM is nevertheless ideally suited to support work in-between meetings, thereby reducing the frequency of meetings.

A new version of VIP Analysis was built as a Decision Deck plug-in. It was built from scratch, since Decision Deck is programmed in Java and is platform-independent, whereas the original version had been programmed in Pascal using Borland Delphi, targeted for Microsoft Windows.
systems only. Moreover, the access to data had to be completely changed from text files based storage to database storage. This was seen as an opportunity to change some of the features of VIP Analysis, abandoning some of the less used ones and introducing some new capabilities.

This paper describes the result of this integration process in Decision Deck: the VIP Analysis Plug-in. The basic decision-aiding concepts of VIP Analysis are described in the next section. The target platform, Decision Deck, is briefly presented in the following section. The fourth section describes D³VIP-A, the VIP Analysis plug-in developed for Decision Deck, with an illustrative case. The concluding section discusses the advantages and shortcomings of D³VIP-A, as well as indicating plans for future developments.

Basic VIP Analysis Concepts

This section overviews the VIP Analysis methodology (Dias & Clímaco, 2000). The purpose of VIP Analysis is to support the evaluation of a discrete set of \( m \) alternatives, in order to choose the most preferred one, according to a multiattribute additive value function (Keeney & Raiffa, 1976; Wakker, 1989). According to this model, the performances of each alternative on each criterion are translated into a value scale by means of a value function, possibly nonlinear (e.g., an increase in performance from 10 to 20 on a given criterion can correspond to an increase of value which is greater than the one resulting from an increase in performance from 110 to 120). The global value of an alternative \( a_i \) (\( i=1,\ldots,m \)) is then the sum of its values for the \( n \) criteria \((v_1(a_i),\ldots,v_n(a_i))\), weighted by \( n \) scaling weights \( w=(w_1,\ldots,w_n) \) that indirectly reflect the importance of the criteria:

\[
V(a_j,w) = \sum_{j=1}^{n} w_j v_j(a_i),
\]

with \( \sum_{j=1}^{n} w_j = 1 \) and \( w_j \geq 0 \).

One of the most difficult steps of the decision aiding process is setting the values of the scaling weights, since these parameters will reflect the DM’s values and trade-offs (e.g., how much would you be willing to lose in attribute “cost” to gain one unit in attribute “safety”?).

In cognitive terms the semantic of these parameters may be difficult to understand for the DM. Scaling weights are artefacts used to translate different value scales into a common unit of value, and their magnitude does not represent directly the importance of the criteria: a criterion can be very important and yet the scaling weight associated with it might be rather small if the performances corresponding to the 0 and 1 levels of the value scale are close.
Another known difficulty is that biases related to the way questions are posed can lead to different outcomes of the elicitation of these values (Weber & Borcherding, 1993). For a DM, value judgments are naturally easier to express through words than through numbers. Furthermore, preferences may evolve, as they are often unstable outcomes of unresolved internal conflicts in the DM’s mind. Adding to these fundamental difficulties, other constraints of a more pragmatic nature may be present, e.g., the DM is reluctant to divulge precise parameter values about his/her preferences in public, or the questioning techniques that can be used to elicit the values of the importance parameters may require more time and patience than the DM can spare.

Moreover, we often need to address the concerns of a group of actors, rather than a single DM. The above mentioned difficulties of fixing preference-related parameter values are still present, if not reinforced by the diversity of judgments and subjective perceptions of what is at stake. In such cases, the existence of “hidden agendas” may hinder an open discussion about parameter values. Even in case of consensus, one must be aware of phenomena such as groupthink (Janis, 1972).

To overcome these difficulties, VIP Analysis proposes to advance in the decision process with Variable Interdependent Parameters. This means that instead of requiring precise values for the scaling weights, it can accept intervals or any other linear constraints on these values. For instance a group of DMs may be doubtful about setting \( w_1=0.2 \) and \( w_2=0.1 \) (these precise values) but may find it easy to agree that \( w_1 > w_2 \). This kind of information is often designated as poor, imprecise, incomplete, or partial information (e.g., see (Dias & Clímaco, 2000; Weber, 1987; Salo & Hämäläinen, 2001)). The constraints usually stem from imprecise answers from the DM (e.g. providing an interval for the trade-off rate between two criteria) or from holistic judgments about alternatives that the DM is able to compare (e.g. \( a_1 \) is preferred to \( a_2 \)).

Let \( W \) denote the set of all combinations (vectors) of parameter values \((w_1,...,w_n)\) that satisfy all the provided constraints at a given moment. Once \( W \) is defined, VIP Analysis may be used to discover robust conclusions (those that hold for every combination in \( W \)) and to identify which results are more affected by the imprecision in the parameter values (the results that vary more).

VIP Analysis can be seen as a toolbox offering complementary approaches to analyze a decision situation with imprecise information. The results produced by VIP Analysis from a set \( W \) of combinations of values for the importance parameters and a set \( A=\{a_1,...,a_m\} \) of alternatives include the following:

a) Computation of a range of value for each alternative \( a_i \in A \): the minimum value of \( a_i \) given \( W \) can be computed by solving a linear program (LP) with the scaling weights \( w=(w_1,...,w_n) \) as variables
\[ \min \{V(a_i, w) : w \in W\}, \]  
(2)

and similarly the maximum value of \( a_i \) given \( W \) can be computed by solving another LP

\[ \max \{V(a_i, w) : w \in W\}. \]  
(3)

If the maximum value for an alternative \( a_i \) is less than the minimum value for an alternative \( a_x \), then the first alternative could be discarded as the second one is clearly superior. The minimum value may be used as a ranking rule – the “maximin” rule (e.g., (Salo and Hämäläinen, 2001)).

b) Computation of the highest difference of value for each ordered pair of alternatives: given an ordered pair of alternatives \((a_i, a_j) \in A^2\) and \( W \), an LP may be solved to find the maximum possible advantage of the first alternative over the second one

\[ m_{ij} = \max \{V(a_i, w) - V(a_j, w) : w \in W\}. \]  
(4)

If the maximum difference is negative or null then \( V(a_j, w) \geq V(a_i, w) \) \( \forall w \in W \), which we denote as \( a_j \Delta a_i \) (\( a_j \) “dominates” \( a_i \)). This means that alternative \( a_j \) has equal or higher value when compared with \( a_i \), for all acceptable weight values. If the maximum difference does not exceed a tolerance parameter \( \varepsilon \), then \( V(a_j, w) \geq V(a_i, w) - \varepsilon \) \( \forall w \in W \), and we denote this as \( a_j \Delta_{\varepsilon} a_i \) (\( a_j \) “quasi-dominates” \( a_i \) with tolerance \( \varepsilon \)). This means that \( a_j \) either wins over \( a_i \), or loses by a negligible difference of value.

c) Computation of the “maximum regret” associated with choosing each alternative: given an alternative \( a_i \in A \), the set \( A \setminus \{a_i\} \), and \( W \), this amounts to find the maximum difference of value by which \( a_i \) can lose to another alternative in \( A \setminus \{a_i\} \). The scaling weights \( w=(w_1, \ldots, w_n) \) are considered as variables (rather than being fixed) to allow the regret to be as high as possible given \( A \) and \( W \)

\[
\text{Regret}_{\max}(a_i) = \max_{w \in W} \left\{ \max_{j \neq i} \left\{ V(a_j, w) - V(a_i, w) \right\} \right\}
\]  
(5)

Rather than directly computing (5), after finding the maximal differences of value (4) for all pairs of alternatives, the maximum regret associated with choosing each one can be found by noting that (Dias & Clímaco, 2000): 

\[
\text{Regret}_{\max}(a_i) = \max_{j \neq i} \left\{ m_{ji} \right\}
\]  
(6)

If \( \text{Regret}_{\max}(a_i) \leq 0 \) then we can say that \( a_i \) is “optimal”; if \( \text{Regret}_{\max}(a_i) \leq \varepsilon \) we can say that \( a_i \) is “quasi-optimal” with tolerance \( \varepsilon \). The “minimax regret” rule can also be used to rank alternatives (e.g., (Salo & Hämäläinen, 2001)), although it is well known for not respecting
Arrow’s independence condition (adding or removing an alternative to/from A may result in rank reversals among the other alternatives).

The outputs of VIP Analysis may allow the decision process to progress, as DMs learn about the model and the problem, postponing the elicitation questions they find difficult to answer, or prone to generate conflict. For instance, they may not agree on precise values for the scaling weights, but may agree on a ranking of those weights. Or they may just agree on a partial ranking (e.g. \( w_1 \geq w_2, \ w_1 \geq w_3, \) and \( w_3 \geq w_4 \)). In any case, VIP Analysis provides results that may allow to eliminate some alternatives or to highlight the most promising ones. Furthermore, the results can be used to direct the elicitation of further information with the purpose of progressively reducing the imprecision (e.g. “can alternative \( a_i \) really attain such a high value?”, “can all DMs agree that \( a_i \) is worse than \( a_j \)?”). The methodology is hence to reiterate several rounds of analysis and discussion of the results interacting with the software.

Very often, only a few constraints, for instance a ranking of the scaling weights suffice to clearly indicating one or two alternatives as being potentially the best ones (Dias & Clímaco, 2000; Salo & Hämäläinen, 2001).

A first implementation of VIP Analysis has been programmed in Pascal language (Borland Delphi). It has been offered as a freeware executable file to everyone who has requested it from its authors. The list of requests is now over 150 users long (not counting students from the authors’ university), mostly academics but also from industry and government institutions, from dozens of different countries.

**The Decision Deck Platform**

The Decision Deck (D²) project in its own words “aims at developing a generic decision aid platform composed of modular and interconnected software components” (Decision Deck, 2008a). These software components implement the common functionalities of a large range of multiple criteria decision aiding (MCDA) methods which makes it possible to easily integrate additional methods as plug-ins. The purpose of the D² project is to provide effective tools for decision aiding consultants, for researchers in the field of MCDA, as well as for educational purposes. The whole project is open-source, being available on the SourceForge website (http://www.sourceforge.org/) under the terms of the GNU General Public License (Free Software Foundation, 2008).

The D² platform is an evolution of the EVAL project (a previous project, funded by the Wallonia Region in Belgium), intending to support alternatives evaluation in a multi-criteria and multi-user (multi-expert) context (Bisdorff et al., 2007). In practice, it is based on the concept of a “heavy client”, communicating (mostly asynchronously and securely) with a database server.
(MySQL), where the data of the models and the users are stored, typically on an Internet server. The client is written in Java, and as such is mostly system independent (running both in Windows and Linux).

Since there are a great number of MCDA methods which could be used to provide alternative evaluation support, the platform itself is MCDA method independent, providing just common functionalities (users management, input of criteria, alternatives and evaluations, data presentation and edition). Some common functionalities, although considered are not yet available (like workflow management and session structuring). A further function of the platform is to provide a repository of the decision process (history, group memory), but at the moment support for this is not yet fully functional.

To use the platform, each user must have an account. Associated to each account there are one or more roles (Bisdorff et al., 2007), which describe functions and capabilities of users:

- **Administrator**: manages the software and database, including the creation of users and assignment of roles;
- **Alternatives designer**: proposes some alternatives or actions;
- **Criteria designer**: defines criteria on which alternatives are to be evaluated;
- **Evaluator**: provides evaluation for the alternatives on the multiple criteria;
- **DM**: expresses preferences on alternatives and criteria;
- **Observer**: “silent” stakeholder who gets informed of the other stakeholders’ activity.

Notice that the concept of Alternative, Criteria, Evaluation and Preference must be adequate to the MCDA method used. Some further roles, requiring support from the particular MCDA method to be effective can also be considered in this design, e.g.,

- **Facilitator**: manages the interaction between the stakeholders of the decision process;
- **“Preferences merger”**: meta-DM who aggregates preferences, from several DMs.

Each account / stakeholder may be granted several of the roles at the same time, but must explicitly choose which one to use on each session.

The actual MCDA method support is provided by plug-in code (MCDA method specific). This code is integrated in the client, and in each D² version several plug-ins can be active at the same time, depending on the choices of the client’s user. As the code is distributed under an open-source license, developers are encouraged to provide new MCDA method plug-ins to improve the platform. This implementation via plug-ins allows for the decomposition of MCDA methods in common functional components and component re-use. For previously implemented MCDA
this approach requires the developer to become familiar with the previous source code, and rewrite the code, conforming to good coding practice to facilitate collaborative development. Although a plug-in creation tool is provided in the current version of D² to help create a “MCDA method skeleton” to be filled in, this approach requires a non-trivial amount of work. In fact, to minimize work in getting existing applications to interact with D², a web-component, named D3 is being developed to facilitate such a task using web-services (Decision Deck, 2008b), but it was not yet ready to be used in this project.

The VIP Analysis Plug-in

Roles and functionality

D²VIP-A, the VIP Analysis plug-in developed for Decision Deck, considers three types of roles for its users: Coordinator (encompasses the roles of Administrator, Alternatives designer, and Criteria designer; more than one coordinator may be present in a decision process), Evaluator, and Decision-Maker (DM). This distinction did not exist in the original VIP Analysis software, which assumed a single user.

Before presenting what possibilities are offered to each of the three roles, let us briefly recall the typical usage of the original VIP Analysis. The single user of VIP Analysis (a DM, an analyst, or a representative of a group) would first define the number of criteria and alternatives, then input the value (not the performance) \( v_j(a_i) \) of each alternative \( (i=1,\ldots,m) \) on each criterion \( (j=1,\ldots,n) \), and input the available information about the scaling weights \( w_j \) \( (j=1,\ldots,n) \). Then, results could be computed, saved, and discussed. The analysis could suggest changes to the inputs in a new iteration.

In D²VIP-A, the existence of three types of user roles supports the workflow associated with a decision process and allows multi-user settings, even in simultaneous. A user can even assume more than one role; the options available will change as his or her role changes.

A user with a role of Coordinator can define a new project, choosing the plug-in or plug-ins the will be available from a list of plug-ins implementing different MCDA methods (Figure 1). This allows analysing the same data with more than one method. Let us now suppose that D²VIP-A is the chosen plug-in, or one of the chosen plug-ins. The coordinator is the only type of user who can set the problem’s name, the names of the alternatives, and the names of the criteria, besides defining who are the other users that can work on the problem and each one’s role or roles.
As an illustration, let us consider an example with one coordinator, three evaluators \((e1, e2, \text{and } e3)\) and two DMs \((d1 \text{ and } d2)\). Their task is to choose a warehouse among four alternatives \((a1, a2, a3, \text{ and } a4)\), considering four criteria \((\text{cost, area, location, and building characteristics})\).

After a Coordinator has defined the problem structure and the users, those users granted the role of Evaluator can evaluate the performance of each alternative under each criterion, according to their expertise. An evaluator is not required to evaluate alternatives on all the criteria, hence allowing processes where the evaluation work is divided among different evaluators; for instance, a financial specialist evaluates the alternatives only on the financial criteria, etc. If the user’s only role is Evaluator, then that user will not have access to others’ evaluations. The performances of the alternatives on a criterion can be provided in any type of scale (e.g., dollars, metric tons, or a qualitative scale). Figure 2 displays the evaluations of the three evaluators on the criterion \(\text{cost}\). Grouping these evaluations by alternative allows seeing the divergence of the evaluators about the total costs that would be incurred by choosing each option.

As the additive aggregation model (1) deals with value scales, rather than performances expressed in the criteria’s original scales, it is necessary to translate performance into value according to a value function for each criterion. As a Coordinator or as a DM, a user can define the shape of value functions that nonlinearily translate the performances of the alternatives on the criteria into a value scale reflecting subjective attractiveness of the performance levels. This possibility did not exist in the original version of VIP Analysis, which assumed that the performances of the alternatives had already been converted into value or utility units. Figure 3, for instance, indicates the value functions for the Coordinator and the two DMs for criterion \(\text{area}\). For \(d1\) increases in area bring diminishing marginal value; for \(d2\) value increases linearly with area (in this range); the Coordinator is considering a value function in-between those proposed by \(d1\) and \(d2\). If the Coordinator allows it, then each DM can also set his or her own value functions; otherwise, all the DMs must use the functions defined by the Coordinator.

The next phase of the analysis is performed by users with the role of DM. It consists in computing results about the global value of each alternative (minimum and maximum value), the value differences between pair of alternatives (maximum difference of value), and the maximum regrets, given a set of constraints on the scaling weights, according to (2)-(6).

The performance evaluations that will serve as a basis for the analysis are chosen by the Coordinator or the DM performing the analysis. Indeed, a Coordinator can indicate a weight for each of the evaluators, reflecting different degrees of confidence in the evaluations. This originates a virtual evaluator defined by the Coordinator, whose evaluations for each criterion correspond to the weighted averages of the evaluators that filled in performance values for that criterion. However, as in the case of defining value functions, the Coordinator may waive this
privilege and let each DM define his or her own virtual evaluator. A Coordinator can hence allow each DM to be free to set value functions and to weigh evaluators, or the Coordinator can ensure that all DMs will work with the same value scales and performances. In the situation illustrated in Figure 4, the Coordinator forces all the DMs to consider all evaluators as weighting the same. The result of the conversion of the original scales to value is depicted in Figure 5, for \(d1\) and the Coordinator. The Coordinator can access these values for all the DMs, but each DM only can see the results for his own value function.

The DM is the only role that can specify the method’s preference-related parameters, i.e. the scaling weights \((w_1, \ldots, w_n)\). In VIP Analysis, the DMs are allowed (or in fact encouraged) to avoid indicating precise numerical bounds for the weights. Rather, they can indicate absolute bounds (e.g., \(w_j \geq 0.05\)), trade-off bounds (e.g., \(w_j \leq 2w_k\)), and/or any kind of linear constraint. One of the innovations with respect to the interface of the original VIP Analysis is the possibility of using a table to easily introduce maximum trade-off rates among pairs of criteria. An example of this possibility is depicted in the “Matrix tradeoffs restrictions” part of Figure 6, filled in by \(d1\). The “1”s in the first column state that none of the weights \(w_2\) (for area), \(w_3\) (for location), and \(w_4\) (for building characteristics), can be higher than \(w_1\) (for cost), whereas the “2”s in the first row state that \(w_1\) can not be higher than \(2w_2\), \(2w_3\), or \(2w_4\), i.e., \(d1\) imposes \(w_j/w_1 \in [1,2]\) \((j=2,3,4)\). An extra “1” relating building characteristics and area further imposes that \(w_4/w_2 \leq 1\).

After introducing the inputs, the DM can compute the minimum and maximum value for each alternative, and the pair-wise confrontation values \(m_{ij}\) (Figure 7, referring to \(d1\)), leading to the maximum regret values. Playing with a tolerance value the DM can relax the definitions of dominance and optimality by considering quasi-dominance and quasi-optimality (Dias & Climaco, 2000). The results are presented not only in tabular form (Figure 8, referring to \(d1\)), but also graphically (Figure 9, referring to \(d2\)).

When the DM decides it is appropriate, he or she can save the results, making these available to the Coordinator. If there are multiple users with a DM role, then the Coordinator can compare the results obtained by each of them and promote a discussion to inform another iteration of the process. In this illustration, the Coordinator can see that \(d1\) would easily exclude alternatives \(a1\) and \(a4\) (Figures 7 and 8). Furthermore, a small tolerance would suffice for \(d1\) to exclude \(a3\). On the other hand, \(d2\), who used slightly different value functions and different constraints on the weights, namely stating \(w_j/w_1 \leq 1(j=2,3,4)\) and \(w_j/w_2 \leq 1(j=3,4)\), would easily exclude \(a3\) and \(a4\) (Figure 9). Furthermore, a small tolerance would suffice for \(d2\) to exclude \(a1\). Hence, there is a reasonable degree of agreement between \(d1\) and \(d2\) to choose \(a2\).
Balance of gains and losses

A comparison between $D^2$VIP-A and the original VIP Analysis mono-user software yields advantages and disadvantages for each tool.

The main advantages of brought by the new version $D^2$VIP-A have already been mentioned: it allows multiple users and different user roles defining a workflow process and enabling concurrent and remote access to the data over a network, and it allows the explicit definition of value functions, rather than forcing the users to convert performance scales into value scales before using the tool.

To these advantages, we may add two additional ones related to the inclusion in the Decision deck platform: the integration with other MCDA methods within a coherent interface, allowing users to easily, and being integrated an open-source environment that facilitates collaborative improvement of $D^2$ and its plug-ins.

On the other hand, the domain visualization of optimality regions present in the original VIP Analysis was not implemented in this new version: it worked only for problems with 2 or 3 degrees of freedom and had been implemented mainly for pedagogical purposes. Another feature of the original software that was not deemed worth implementing was the criteria-based activation / deactivation of alternatives: all alternatives are considered as active unless deleted. Nevertheless, there is no technological barrier to implementing these features in the future.

An important difference between the original VIP Analysis and $D^2$VIP-A, which can be seen as an advantage or disadvantage, concerns the way data (inputs and outputs) are stored. The former uses text files, which could be prepared and read using a word processor (or even a spreadsheet). The text files were stored at the user’s computer, needing to be moved if it became necessary to work on a different computer.

$D^2$VIP-A uses a MySQL database, which can be accessed concurrently by many users if located on a network server (possibly connected to the Internet). However, this increased flexibility in access comes at the cost of not being easy to interact with word processors and spreadsheets. Developing $D^3$ plug-ins for importing and exporting data to text files or XML files is nevertheless a way to overcome this limitation in the future.

A final aspect that some users may consider as a disadvantage of the new version is the need to install a MySQL database server, which makes $D^2$VIP-A unwieldy for individual use on a single PC (it is also necessary to install a Java Runtime Environment, but many computers will already have it installed).
Concluding remarks

This paper presented the VIP Analysis plug-in of D². This new implementation maintains the methodology of the original version unchanged, but benefits from the integration in the Decision Deck environment as a multi-method decision support system. In this environment, D²VIP-A is included side by side with other multi-criteria evaluation methods, offering the potential users an integrated and uniform interface. The plug-in strategy of D² also allowed reusing already programmed modules. Another noteworthy benefit of the platform is the independence towards any operative system (whereas the original VIP Analysis only runs on Microsoft Windows). We can also mention the possibility of having multiple users collaborating over a network on different roles attributed by the coordinator.

All these advantages, plus the advantages of being distributed as open source software (thus allowing for customization and collaborative improvement), also apply when comparing this new version with well-known packages for multiatribute additive value functions such as Hiview or Logical Decisions (Weistroffer et al., 2005).

We cannot consider that the old version of VIP Analysis has become obsolete and that it will cease to be distributed, since some users may still prefer it. This may occur due to the differences in implemented features, but these are small; most users that choose to use the older version in the future will do so for its simpler data management (no need to install an SQL server) in single-user situations, and the possibility to easily edit inputs and outputs on a spreadsheet or a word processor.

We can identify some current limitations of D²VIP-A for multi-user collaboration. Firstly, it is poor as a communication channel, requiring other programs (e-mail, irc, or other) for an effective coordination and for informal exchange of information. Secondly, mechanisms provided to support mutual awareness (Gutwin & Greenberg, 2002), such as knowing who is logged in at a given moment, what are those users doing, or what has changed during an user’s absence, are insufficient or non-existing. Some of the feedback mechanisms proposed by (Dias & Clímaco, 2005) have not been implemented due to these limitations.

Future work is underway to overcome these limitations concerning communication and awareness, allowing to fully implementing a Group Decision Support System. Other aspects such as offering workflow support with tools such as timelines and agendas, or facilitating importation / exportation of inputs and outputs from / to other applications are also planned. The future implementation will be based on web-services to allow its integration in D3, as well as to allow simpler browser-based interfaces.
Acknowledgements

The authors gratefully acknowledge the constructive comments of the Editors and two anonymous referees, as well as the technical support and advice from Gilles Dodinet and Michel Zam, from Karmic Software. This work has been partly supported by the FCT / POCI 2010 program grant EGE/58371/2004.

References


Figure 1. Selection of plug-ins to be used when creating a project, among available plug-ins.
Figure 2. Coordinators and DMs can visualize the performances of all evaluators on all criteria (in this case, criterion cost).
Figure 3. Value functions of the Coordinator and the two DMs for criterion *area*.

Figure 4. A Coordinator can set the evaluators’ weights and can force all the DMs to use the same weights.
Figure 5. Value function evaluations of the 4 alternatives for DM d1 and the Coordinator (accessible to the latter).

Figure 6. Constraints on the weights placed by d1.
Figure 7. Maximum differences of value for \( d1 \). The highlighted dominance situations show \( a1 \) and \( a4 \) as dominated alternatives; if a tolerance of 0.08 is accepted then \( a3 \) would be considered quasi-dominated by \( a2 \).

Figure 8. Outputs in tabular form for \( d1 \): maximum value, weights leading to the maximum value, minimum value, weights leading to the minimum value, and maximum regret.
Figure 9. Graphical display of value range for different alternatives, for $d_2$. 